Random Forest

Dave Tang

13 May 2016

Contents

Install required packages

```
required_packages <- c('randomForest', 'ggplot2')

for(p in required_packages){
  if (! p %in% installed.packages()[,1]){
    install.packages(p)
  } else {
    print(paste('The package ', p, ' is already installed', sep=''))
  }
}
```

```
## [1] "The package randomForest is already installed"
## [1] "The package ggplot2 is already installed"
```

Preparing the data

```
data_url <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data'
df <- read.table(file=url(data_url), header=FALSE, sep=",")
header <- c('class',
            'alcohol',
            'malic_acid',
            'ash',
            'ash_alcalinity',
            'magnesium',
            'total_phenols',
            'flavanoids',
            'nonflavanoid_phenols',
            'proanthocyanins',
```

```
            'colour',
            'hue',
            'od280_od315',
            'proline')
names(df) <- header
df$class <- as.factor(df$class)

# how many classes of wine?
table(df$class)
```

```
##
##  1  2  3
## 59 71 48
```

Analysis

```
library(randomForest)
```

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(31)
my_sample <- sort(sample(x = 1:nrow(df), replace = FALSE, size = nrow(df)/2))
my_sample_comp <- setdiff(1:nrow(df), my_sample)

test <- df[my_sample, ]
train <- df[my_sample_comp, ]

# data = an optional data frame containing the variables in the model
# importance = calculate the importance of predictors
# do.trace = give a more verbose output as randomForest is running
# proximity = calculate the proximity measure among the rows
r <- randomForest(class ~ ., data=train, importance=TRUE, do.trace=100, proximity = TRUE)
```

```
## ntree      OOB      1      2      3
##   100:   4.49%  0.00% 11.11%  0.00%
##   200:   1.12%  0.00%  2.78%  0.00%
##   300:   2.25%  0.00%  5.56%  0.00%
##   400:   1.12%  0.00%  2.78%  0.00%
##   500:   1.12%  0.00%  2.78%  0.00%
```

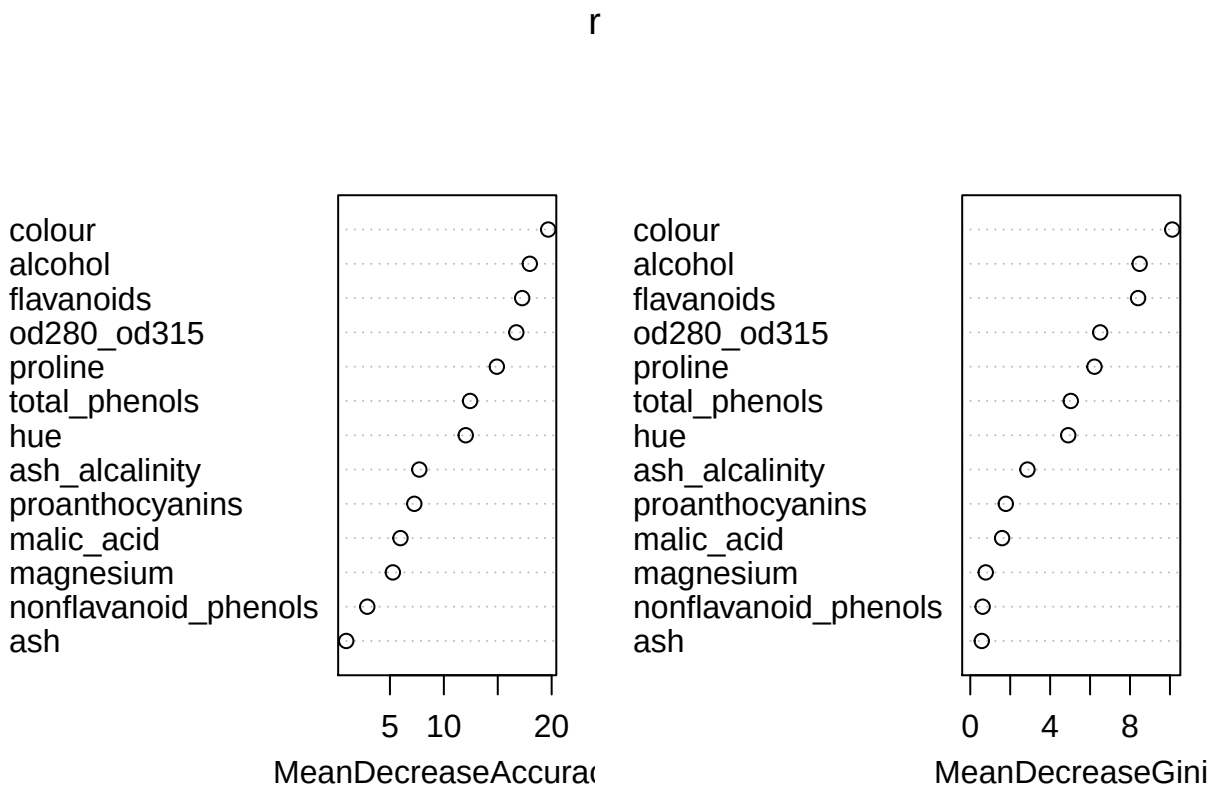
Plots

```
library(ggplot2)
```

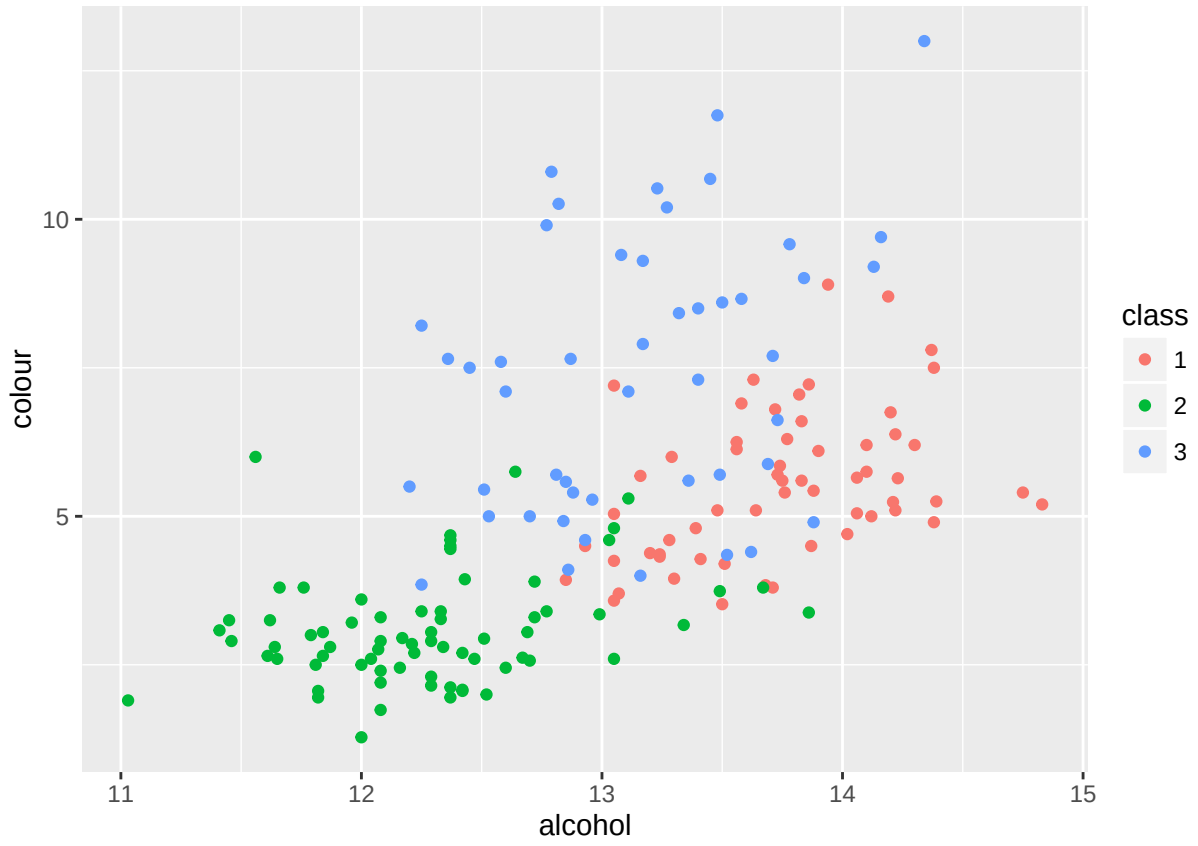
```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
varImpPlot(r)
```




```
ggplot(df, aes(x=alcohol, y=colour, colour=class)) + geom_point()
```



Random Forest object

```
class(r)
```

```
## [1] "randomForest.formula" "randomForest"
```

```
names(r)
```

```
##  [1] "call"            "type"            "predicted"
##  [4] "err.rate"        "confusion"       "votes"
##  [7] "oob.times"       "classes"         "importance"
## [10] "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"
## [16] "y"               "test"            "inbag"
## [19] "terms"
```

```
# the original call to randomForest
r$call
```

```
## randomForest(formula = class ~ ., data = train, importance = TRUE,
##     do.trace = 100, proximity = TRUE)
```

```
# one of regression, classification, or unsupervised
r$type
```

```
## [1] "classification"
```

```
# the predicted values of the input data based on out-of-bag samples
r$predicted
```

```
##   1   2   5   6   9  10  13  16  24  26  28  29  32  34  36  37  39  40
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  43  44  45  46  48  50  51  53  54  55  57  65  66  69  70  72  74  78
##   1   1   1   1   1   1   1   1   1   1   1   2   2   2   2   2   1   2
##  81  83  85  86  87  90  91  95  96  97  98 100 101 104 105 106 107 108
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
## 111 113 114 117 118 120 122 123 126 128 129 131 132 133 137 140 141 142
##   2   2   2   2   2   2   2   2   2   2   2   3   3   3   3   3   3   3
## 143 146 149 151 152 155 157 161 162 164 166 167 168 172 173 176 177
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## Levels: 1 2 3
```

```
# a matrix with nclass + 2 (for classification) or two (for regression) columns
# for classification: the first three columns are the class-specific measures
# computed as mean descrease in accuracy
# the MeanDecreaseAccuracy column is the mean descrease in accuracy over all classes
# the MeanDecreaseGini is the mean decrease in Gini index
r$importance
```

```
##                                1            2           3
## alcohol               0.1400866328  0.1246802680 0.0310179931
## malic_acid            0.0072780747  0.0002262662 0.0210402819
## ash                  -0.0002868465  0.0001398091 0.0021564935
## ash_alcalinity        0.0187454962  0.0049117013 0.0330122655
## magnesium             0.0123535687  0.0003245882 0.0009802309
## total_phenols         0.0887444232  0.0055303446 0.1001674881
## flavanoids            0.1073962534  0.0322769633 0.1668975247
## nonflavanoid_phenols  0.0044838772 -0.0010135314 0.0027388889
## proanthocyanins       0.0066483508  0.0033661938 0.0327330170
## colour                0.1383499448  0.1486293740 0.1173514819
## hue                   0.0371697441  0.0176351345 0.0747928460
## od280_od315           0.0903056670  0.0344418438 0.1377274059
## proline               0.1342962400  0.0393595001 0.0267130092
##                      MeanDecreaseAccuracy MeanDecreaseGini
## alcohol                        0.1041064093        8.4820578
## malic_acid                     0.0077632411        1.5955392
## ash                            0.0005483401        0.5770704
## ash_alcalinity                 0.0161728508        2.8642512
## magnesium                      0.0043882588        0.7747925
## total_phenols                  0.0573982757        5.0339269
## flavanoids                     0.0895104064        8.4047643
## nonflavanoid_phenols           0.0021143766        0.6187621
## proanthocyanins                0.0119130913        1.7802857
## colour                         0.1347744714       10.1153258
```

```
## hue                           0.0382458993        4.9056291
## od280_od315                   0.0785702671        6.5071186
## proline                       0.0664069780        6.2196224
```

```
# the "standard errors" of the permutation-based importance measure
r$importanceSD
```

```
##                                1           2           3
## alcohol              0.008582197 0.0081250438 0.0042473846
## malic_acid           0.002111788 0.0010774274 0.0039032287
## ash                  0.001082018 0.0007792105 0.0011955269
## ash_alcalinity       0.003283561 0.0021549052 0.0046081398
## magnesium            0.002378783 0.0008080027 0.0009632592
## total_phenols        0.008110053 0.0019957919 0.0086667421
## flavanoids           0.008255754 0.0038564467 0.0112153212
## nonflavanoid_phenols 0.001633503 0.0010833690 0.0012175023
## proanthocyanins      0.002423741 0.0017216863 0.0044485576
## colour               0.008410772 0.0086560781 0.0087340585
## hue                  0.004223975 0.0026889316 0.0074020645
## od280_od315          0.007412297 0.0037121930 0.0097885138
## proline              0.009302794 0.0045853358 0.0044194205
##                      MeanDecreaseAccuracy
## alcohol                        0.0057932387
## malic_acid                     0.0012997619
## ash                            0.0005811038
## ash_alcalinity                 0.0020940327
## magnesium                      0.0008330144
## total_phenols                  0.0046166533
## flavanoids                     0.0051843834
## nonflavanoid_phenols           0.0007302978
## proanthocyanins                0.0016410108
## colour                         0.0068481647
## hue                            0.0031792664
## od280_od315                    0.0046985461
## proline                        0.0044524568
```

```
# number of trees grown
r$ntree
```

```
## [1] 500
```

```
# number of predictors sampled for spliting at each node
r$mtry
```

```
## [1] 3
```

```
# a list that contains the entire forest
# r$forest
# use getTree() to obtain an individual tree
getTree(r, k = 1)
```

```
##    left daughter right daughter split var split point status prediction
## 1              2              3         7       0.955      1          0
## 2              0              0         0       0.000     -1          3
## 3              4              5         7       2.565      1          0
## 4              6              7         1      12.605      1          0
## 5              8              9         1      13.020      1          0
## 6              0              0         0       0.000     -1          2
## 7             10             11        12       1.700      1          0
## 8              0              0         0       0.000     -1          2
## 9              0              0         0       0.000     -1          1
## 10             0              0         0       0.000     -1          3
## 11            12             13        10       3.560      1          0
## 12             0              0         0       0.000     -1          2
## 13             0              0         0       0.000     -1          1
```

```
# (classification only) vector error rates of the prediction on the input data,
# the i-th element being the (OOB) error rate for all trees up to the i-th
head(r$err.rate)
```

```
##             OOB          1          2          3
## [1,] 0.03030303 0.00000000 0.07142857 0.00000000
## [2,] 0.07547170 0.05555556 0.08695652 0.08333333
## [3,] 0.10294118 0.08695652 0.13333333 0.06666667
## [4,] 0.08219178 0.08000000 0.12903226 0.00000000
## [5,] 0.11111111 0.07692308 0.15151515 0.09090909
## [6,] 0.10714286 0.07407407 0.17647059 0.04347826
```

```
# (classification only) the confusion matrix of the prediction (based on OOB data)
r$confusion
```

```
##    1  2  3 class.error
## 1 29  0  0  0.00000000
## 2  1 35  0  0.02777778
## 3  0  0 24  0.00000000
```

```
# (classification only) a matrix with one row for each input data point and one column
# for each class, giving the fraction or number of (OOB) 'votes' from the random forest
head(r$votes)
```

```
##            1           2          3
## 1  0.9735450 0.026455026 0.00000000
## 2  0.9682540 0.021164021 0.01058201
## 5  0.7142857 0.217142857 0.06857143
## 6  0.9831461 0.016853933 0.00000000
## 9  0.9946524 0.005347594 0.00000000
## 10 0.9513514 0.048648649 0.00000000
```

```
# number of times cases are 'out-of-bag' (and thus used in computing OOB error estimate)
r$oob.times
```

```
##  [1] 189 189 175 178 187 185 195 205 158 180 188 209 185 193 180 187 184
```

```
## [18] 188 193 162 201 177 186 189 169 159 185 162 185 200 176 192 192 176
## [35] 184 182 181 197 169 169 183 192 195 190 197 186 173 182 172 200 188
## [52] 212 169 181 172 178 204 191 176 194 192 180 182 162 177 175 180 172
## [69] 176 189 192 174 182 148 187 185 165 198 188 174 197 169 183 172 189
## [86] 185 186 192 189
```

```
# if proximity=TRUE when randomForest is called, a matrix of proximity measures among the
# input (based on the frequency that pairs of data points are in the same terminal nodes)
dim(r$proximity)
```

```
## [1] 89 89
```

On importance

Notes from Stack Exchange:

MeanDecreaseGini is a measure of variable importance based on the Gini impurity index used for the calculation of splits during training. A common misconception is that the variable importance metric refers to the Gini used for asserting model performance which is closely related to AUC, but this is wrong. Here is the explanation from the randomForest package written by Breiman and Cutler:

> Every time a split of a node is made on variable m the gini impurity criterion for the two descendent nodes is less than the parent node. Adding up the gini decreases for each individual variable over all trees in the forest gives a fast variable importance that is often very consistent with the permutation importance measure.

The Gini impurity index is defined as:

$$G = \sum_{i=1}^{n_c} p_i(1 - p_i)$$

where n_c is the number of classes in the target variable and p_i is the ratio of this class.

Session information

```
## R version 3.2.4 (2016-03-10)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.5 (Yosemite)
##
## locale:
## [1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggplot2_2.1.0       randomForest_4.6-12
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.4      digest_0.6.9     plyr_1.8.3       grid_3.2.4
##  [5] gtable_0.2.0     formatR_1.3      magrittr_1.5     evaluate_0.8.3
```

```
##  [9] scales_0.4.0     stringi_1.0-1    rmarkdown_0.9.5  labeling_0.3
## [13] tools_3.2.4      stringr_1.0.0    munsell_0.4.3    yaml_2.1.13
## [17] colorspace_1.2-6 htmltools_0.3.5  knitr_1.12.3
```